deepinstinct™
BEFORE YOU KNOW IT

MAKING SENSE OF FILELESS MALWARE

March 2018

TABLE OF CONTENT

INTRODUCTION

Threat actors are constantly searching for new and sophisticated ways in which to avoid detection in order to successfully perform malicious attacks. The increasing power and ambition of threat actors was especially evident during 2017 – when an estimated 230,000 malware samples were produced daily, and 4,000 ransomware attacks threatened organizations daily.[1] Apart from the rise in ransomware attacks, 2017 also saw an extreme increase in the amount of fileless malware attacks, which grew by approximately 50% in 2017. [2]

Fileless malware attacks pose a threat to organizations and a challenge for security vendors, due to the use of various non-executable file formats for infection, and the ability to execute parts of the malicious logic of the attack in-memory only. Several high-profile groups, such as FIN7, Lazarus, and Oilrig performed damaging fileless attacks during 2017. This phenomenon has encouraged cyber criminals to more extensively adopt fileless attack techniques, which were once mostly used by nation-states, leading to a rise in the amount of fileless attacks. This whitepaper will present a comprehensive overview of fileless malware, enabling individuals and organizations to better understand and deal with the threat posed by fileless malware.

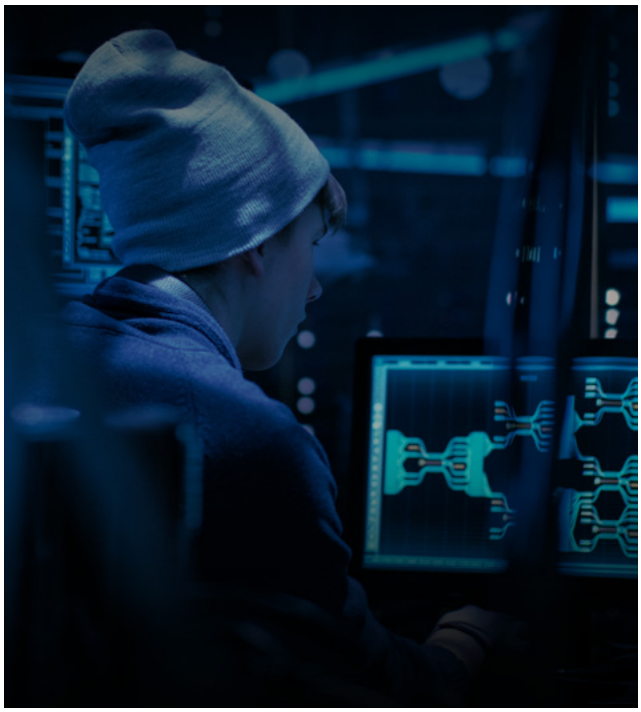
IN THIS WHITEPAPER WE WILL COVER:

- What fileless malware is, and how it differs from other types of malware
- A deep dive into the attack methods used in a fileless malware attack, including detailed examples for each method
- An overview of how the security industry is dealing with fileless malware
- Ways in which organizations and individuals can protect themselves from fileless malware
- Trends and predictions on the future of fileless malware

FILELESS MALWARE

AN OVERVIEW

Threat actors are constantly developing and searching for new ways to avoid detection. Fileless attacks, which technically exist since 2001[*3], are an example of such a method. The definition of what is considered fileless malware is wide, as the term "fileless malware" encompasses several possible attack scenarios, only some of which don't write any files to disk, while very few scenarios are completely fileless. A widely accepted definition of a fileless attack is an attack during which no portable executable (PE) file is written to and executed from disk.



THE FOLLOWING ATTACK SCENARIOS FALL UNDER THE SCOPE OF THE CURRENT ACCEPTED DEFINITION OF A FILELESS ATTACK:

Executable-less attacks: attacks based on a dropper, usually a document or zip file,which runs a script that executes the next stages of the attack. This attack is file based, since the dropper is saved to disk. However, executables are usually not used during the whole attack sequence, so this type of attack is better termed an "executable-less attack". This attack scenario is the most common "fileless" scenario.

Dual use attacks: attacks based on legitimate files which are either common to the organization attacked or are widely-used administrative tools, which can be abused to perform malicious functions. This attack scenario is also called "living off the land", and involves executable files.

Code injection attacks: attacks based on code injection into the memory of a process. In this attack, the malicious logic is not saved to disk, however the attack usually includes files, since the injection to memory is usually performed by script files, which are saved to disk.

Memory-only attacks: advanced attacks which occur only in memory, and do not write any files to disk during the whole attack sequence. These attacks are extremely advanced, and are currently performed mostly by nation-state actors.

* For more details, please see note 3 in Bibliography

ATTACK SCENARIOS

CATEGORY	INVOLVES FILES	FILE TYPES INVOLVED	MEMORY INJECTION	RECENT EXAMPLES
Script based: "Executable-less" attacks	✓	▪ Document droppers (PDF, DOC...) ▪ Scripts (VBS, PowerShell, JS...)	✗	Cobalt malware used a document dropper which launched a chain of several scripts.
Dual-use: "Living off the land" Attacks	✓	Portable executables	✗	Windows Sysinternals, netsh, Mimikatz
Code injection attacks	✓	Scripts and/ or droppers to inject the code.	✓	The Poweliks Trojan injects its DLL into one of several processes to run after system start-up.
Memory-only attacks	✗		✓	Targeted attack against financial institutions, attributed to the Lazarus group.

As can be seen above, a more accurate term for "fileless attack" is "executable-less attack", since, in most scenarios of so-called fileless attacks, files which are not portable executables, such as various documents or script files, are written to disk. Moreover, in many attack sequences PE files may be used, but not written to disk, while in some attack sequences no PE files are used during the whole attack.

According to a recent survey conducted by the Ponemon Institute, it was estimated that 20% of attacks on organizations during 2016 were fileless attacks - a number which has grown to 29% in 2017, and is estimated to increase by 35% in 2018. Furthermore, 42% of survey respondents say their companies experienced at least one successful fileless attack in 2017, while 75% of all successful compromises involved fileless methods[4]. A survey by SANS Institute provided similar results - showing almost one-third of respondents experienced a fileless threat entering their organization during 2017[5]. In addition, a December 2017 threat report stated that 52% of all attacks in 2017 were fileless malware attacks utilizing PowerShell or Windows Management Instrumentation (WMI)[6]. These findings highlight the threat that is posed by fileless malware to organizations, a threat which is growing every year.

THE ANATOMY OF A FILELESS ATTACK

This section will review the anatomy of a fileless attack. It will cover different techniques used for infection, malicious logic, lateral movement, and persistence, and will provide examples for each technique. It should be noted that most of the techniques which will be covered are not unique to fileless attacks, but in the context of a fileless attack these techniques could pose a greater risk to attacked organizations and individuals, as they could make the attack more covert.

This section is based on research conducted on a dataset of dozens of known fileless campaigns and malware samples. During this research, the main courses of action of fileless malware were identified and analyzed, and examples for each course of action were collected, and are presented in the next section.

INFECTION

A fileless attack sequence begins with initial infection of the target, which can be achieved through one of the following techniques:

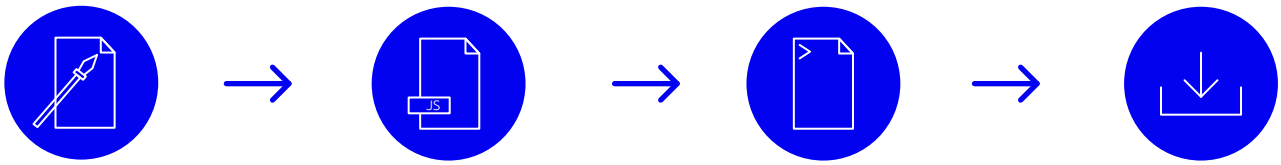
DROPPERS

Involves files: ☑

Possible attack scenarios:
executable-less, code injection

Similarly to other types of attacks, in a fileless malware attack, initial hold of the victim can be achieved through a phishing attack which contains a dropper - such as a PDF, RTF, Office file, or archive. In most cases, the dropper will then run a script, which can be either a VBA macro, or another type of script, such as PowerShell (PS), JS or HTA. In some cases, the scripts act as downloaders, either downloading a PE file to disk before removing it, injecting a PE file into another process, or downloading another script to carry out the next stage of the attack. In rare cases, the entire malicious logic is contained within the script. In other cases, vulnerabilities in the document reader, for example Adobe Acrobat, can be exploited to drop the next phase of the attack.

Example: An attack using this method was the Cobalt malware, which used a document dropper exploiting CVE-2017-11882. When the document was opened by the user, the exploit contained in the document downloaded a JavaScript, which in turn executed several PowerShell scripts – the last of which contained Cobalt DLL's in the script code, which were executed in PowerShell's memory, without being dumped to disk.[7] Through use of this exploit, attackers were able to execute a fileless attack, in which the only action performed by the user was opening the document dropper.



spearphishing doc → doc downloads JavaScript → JavaScript executes PS → PS downloads shellcode and injects it

Based on our analysis, the vast majority of fileless campaigns use droppers in order to gain an initial foothold on a victim. The use of droppers is very common not only in fileless malware, but also in file-based malware attacks. Many well-known ransomware and financial malware campaigns, such as the Spora ransomware or the Corebot banking trojan utilize this method.

INJECTION FROM THE BROWSER

Involves files ☒

Possible attack scenarios
executable-less, code injection, memory-only

The attack is initiated through exploitation of a browser vulnerability, in order to run a script that collects information on the victim and downloads the next stage of the attack, which can be a fileless or a regular payload. One tool which can carry out this form of attack are exploit kits. Exploit kits are tosolkits that combine exploits to attack vulnerable clients. Generally, exploit kits contain browser related exploits, such as Adobe Flash, Internet Explorer or Microsoft Silverlight exploits. Such kits were once very popular and effective among cyber-criminals, and were used to attack victims mainly through browser vulnerabilities. However, successful infections by exploit kits have greatly decreased, mainly due to improved browser security, leading to a lack of unpatched exploits. All recent successful infections by exploit kits are due to unpatched software, as exploit kits do not exploit zero-day vulnerabilities.

Example: In August 2014 the Angler Exploit Kit used an exploit to inject the downloaded malware directly into the browser process which was being exploited. Through this method, Angler was able to run many types of malware directly in-memory, including the fileless Poweliks malware – resulting in a completely fileless attack chain[8].

Infection exploiting the browser make both detection and post-event analysis very difficult, since the malware exists only in the memory of the infected host. However, due to the increasing difficulty in finding browser vulnerabilities and successfully exploiting them, this method of attack is in decline.

MALICIOUS LOGIC AND LATERAL MOVEMENT

Like other types of malware, payload delivery and lateral movement will usually follow a successful infection. The payload will perform actions desired by the attacker, such as information collection, file encryption, or backdoor communication, while lateral movement can be used to infect additional computers within the network.

SCRIPTS

Involves files ☑

Possible attack scenarios
executable-less,
code injection

Scripts are an increasingly used filetype in fileless attacks. It's important to note that since script files are usually saved to disk, the better terminology for this attack is "executable-less" attacks. The use of scripts poses many advantages to the attacker: scripts are easy to write and execute, trivial to obfuscate, and can be extremely polymorphic. In addition, many types of script files can be used in attacks – the most popular being PowerShell, JS, HTA, VBA, VBS, and batch scripts. These scripts can be run on almost all Windows systems, increasing the chances of successful infection. One major drawback of this method is that for the script to run, user interaction is required, and warning windows are presented to the user, unless an exploit is used. For example, in most cases the script is contained either as a script file within an email, which the user can then run by clicking the script; or as a VBA macro in a document, for which the user needs to enable macros. As mentioned earlier, scripts can also be used during initial infection of a victim.

This method is used by many types of malware, including ransomware and backdoors, and is not limited to fileless attacks, as a script which downloads a PE file can either save it to disk or run it from memory, depending on its level of sophistication. The script can also perform additional malicious functionalities, such as collecting information about the victim, from the computer name to saved passwords. 75% of fileless campaigns analyzed while working on this whitepaper use scripts (mostly one or more of PS, HTA, JS, VBA) during at least one of the attack stages.

Example: The Helminth Trojan, used by the Iran-based Oilrig group, uses scripts for its malicious logic. In the attack, a Word Document file exploiting CVE-2017-0199 delivers an HTA script, which is executed by the Windows process which executes HTML executables, mshta.exe. Once executed, the script initiates the attack, which delivers the Helminth Trojan as PowerShell and VBS files[9].

NATIVE TOOLS

Involves files ☑

Possible attack scenarios
executable-less, dual-use, code-injection, memory-only

The Windows operating system contains several native Windows tools (tools which are installed by default with the Windows operating system) which are used prominently in fileless attacks. These tools include Interactive PowerShell (first introduced in WindowsXP and Vista) and Windows Management Instrumentation (WMI, first introduced in Windows 2000). These tools are heavily used in fileless and non-fileless attacks, as they can both be used by malicious actors for persistence, malicious logic, and lateral movement. The inclusion of both tools as native components of Windows had a dramatic influence on their usage, both by system administrators and malicious actors

Interactive PowerShell: is a .NET based Windows command-line shell designed especially for system administrators, and includes an interactive prompt and a scripting environment, and can run .NET functions. The PowerShell interface provides access to the file system on the computer, and enables access to other data stores, such as the registry and the digital signature certificate stores[10]. Due to this access, PowerShell can be used by malicious actors for a variety of purposes. In addition, there are open source tools kits which can assist attackers in fileless attacks – such as Powersploit, a PowerShell post exploitation framework[11].

WMI: another native Windows tool, is the infrastructure for management data and operations on Windows operating systems. WMI scripts or applications can be used to automate administrative tasks on local or remote computers[12] , and are in wide use in organizational networks. Due to its power, malicious actors familiar with WMI can use it for many purposes, such as persistence, information collection, and lateral movement. Malware can even be created in such a way that it performs its malicious logic using only WMI. This type of malware can be very hard to detect, as its actions are only performed in-memory, through a legitimate, much used system tool. WMI can be interacted with in several ways, including through PowerShell or other scripting languages, such as JS. As WMI can perform reconnaissance, and create processes (through the Win32_Process Create method), it can be used to perform a completely fileless attack – malware can be delivered through a document which will invoke PowerShell, which will then interact with WMI to perform the malicious logic.[13]

Example: APT29, a nation state actor, uses WMI in one of its backdoors both as a means of persistence, and to store the PowerShell code of the backdoor.[14]

Overall, 30% of the fileless campaigns we analyzed utilize WMI.

LIVING OFF THE LAND

Involves files ☑

Possible attack scenarios
dual-use,
code-injection

A rising trend not only in fileless attacks, but in malware attacks in general, is called "living off the land". Living off the land refers to the use of dual-use tools - which are either already installed in the victims' environment, or are admin, forensic or system tools used maliciously. Examples for dual-use tools which have been used for "living off the land" attacks are the Windows Sysinternals, NETSH or SC tools, or forensic tools such as the password extracting tool Mimikatz, or other tools such as SSH which could be used as backdoor. In some fileless attacks, these tools are downloaded by the malware and saved to disk – however the attack is still commonly considered fileless in this case, as these tools have legitimate uses, and so would not be detected by most security vendors. In other cases, these tools are injected into memory and not saved to disk.

According to a report published by Symantec in July, all recent advanced nation-state sponsored attacks used dual-use tools.[15]

Example: In February, a wide-spread fileless attack against financial institutions around the world was reported. The attackers deployed several tools to gain control of the financial networks. To gain administrative privileges, the attackers used Mimikatz for password collection. Then, the attackers placed PowerShell scripts in the registry, and generated Metasploit scripts that were executed using the Windows SC service. In addition, the Windows NETSH tool was used for communication between the victim and the C2[16]. This comprehensive use of tools enabled the attackers to perform a highly sophisticated attack, which remained undetected for some time.

INJECTION TO MEMORY

Involves files ☑

Possible attack scenarios
code-injection, memory-only

To execute PE code without writing files to disk, attackers can use one of several methods to inject that code into memory, enabling the code to execute in-memory. For example, attackers can inject their code into currently running legitimate processes, or create a new instance of a legitimate process and inject their code to it. Many methods exist to achieve this, the most recent of which is called Process Doppleganging. This technique makes use of NTFS transactions to modify an executable, which is then loaded in a process, while the changes in the executable are rolled back. Thus, a process is created with the modified executable, while the actual executable appears unmodified.[17] As no files are saved to disk during Process Doppleganging, or during other similar methods, these methods can be detected only by in-memory scanning or heuristics. Additional techniques include reflective DLL or EXE injection, classic DLL injection, or process hollowing.

Example: The Poweliks Trojan underwent several stages before launching a fileless variant in 2015. The Trojan, which is a widespread click-fraud malware, uses code injection as part of its attack sequence. As part of its attack, Poweliks uses a DLL in order to ensure persistence, and communicate with the C2. Through COM hijacking (COM hijacking is reviewed in more detail in the persistence section), Poweliks ensures that the registry key containing the DLL is run at each system start-up, launching one of several legitimate processes (such as regsvr32.exe), and injecting the DLL into them.[18]

According to our analysis, 50% of fileless attacks perform code injection during at least one stage of their attack.

PERSISTENCE

Achieving persistence is a considerable challenge for fileless malware, since executable files performing the attack are usually not saved to disk. Attackers that wish to remain persistent must leave some trace in the system for their malicious logic to relaunch itself following system startup. There are also attackers which do not seek persistence, such as ransomware authors, which only need to run their malware once for it to achieve its goals.

To remain persistent, attackers can use one of several means. These means are usually achieved through scripts involved in the attack process, and can be one of the following:

REGISTRY EDITING

Involves files ☑

Possible attack scenarios
executable-less, dual-use, code-injection

The Windows Registry is a hierarchically organized database that stores settings for the Windows operating system, and for applications installed on the system. There are many ways in which registry modification can allow malware to remain persistent on a system, for example through editing of the Run/RunOnce keys, which lists what programs run when a user logs on.

Example: Poweliks, mentioned above, creates a load point in the registry, which it then uses in order to inject itself into a process, and begin performing its malicious click-fraud functionality.[19] Registry editing is widely used as a means of persistence, as 30% of fileless malware analyzed for this paper uses this method to gain persistence.

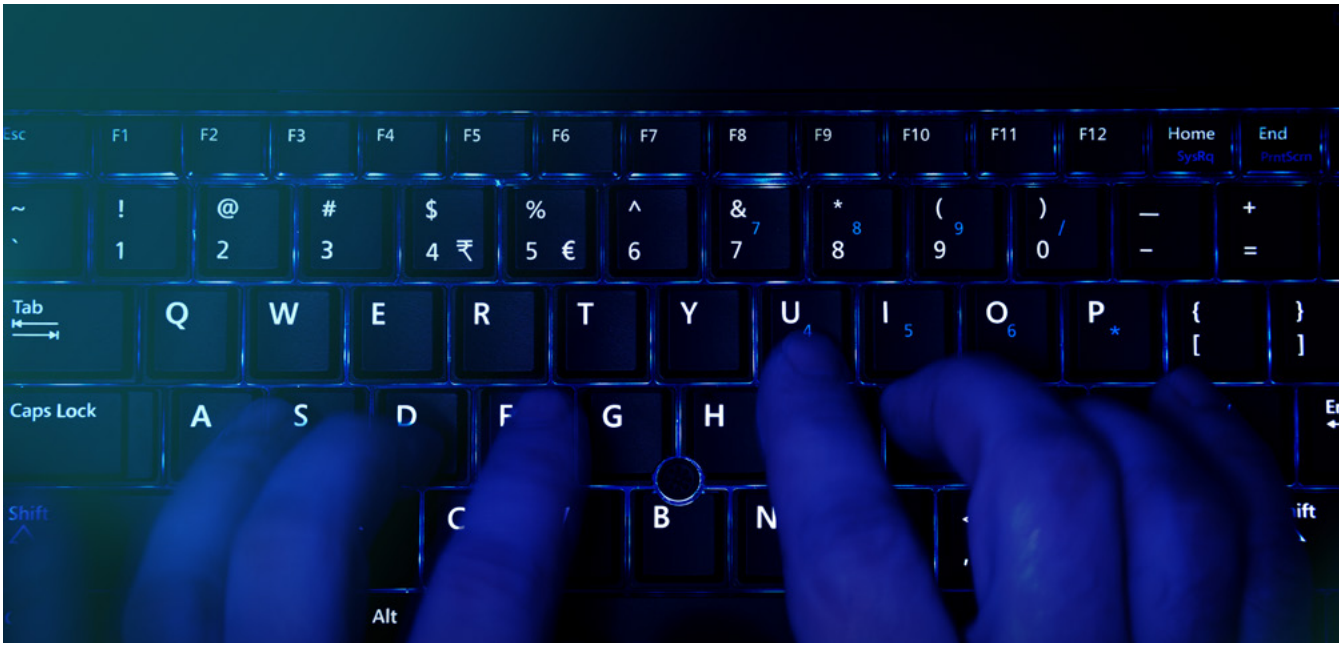
COM HIJACKING

Involves files ☑

Possible attack scenarios
executable-less, dual-use, code-injection

The Microsoft Component Object Model (COM) is a system within Windows which enables interaction between software components through the operating system.[20] Windows components, some of which are automatically loaded at system startup, have a unique COM identifier, called a CLSID, which is located in the Windows Registry. Malicious actors can hijack the registry reference to the CLSID, and place malicious code which will be executed when the COM object is executed.[21]

Example: The COMpfun RAT, uncovered in 2014, uses COM hijacking as a means of persistence. The RAT places the contents of its 32bit and 64bit payloads under the CLSID's b5f8350b-0548-48b1-a6ee-88bd00b4a5e7 and BCDE0395-E52F-467C-8E3D-C4579291692E. [22]



CREATION OF SCHEDULED TASKS

Involves files ☑

Possible attack scenarios
executable-less, dual-use, code-injection

The Windows Task Scheduler is a native Windows component that provides users with the ability to schedule various tasks, such as the execution of programs or scripts. This tool is widely used for legitimate uses, and can also be abused by malware. Malware can abuse the built in Scheduled Tasks tool to create their own malicious tasks for persistence or information collection.

Example: The Helminth Trojan used a scheduled task to execute one of its components, a malicious VBS, every 3 minutes.[23]

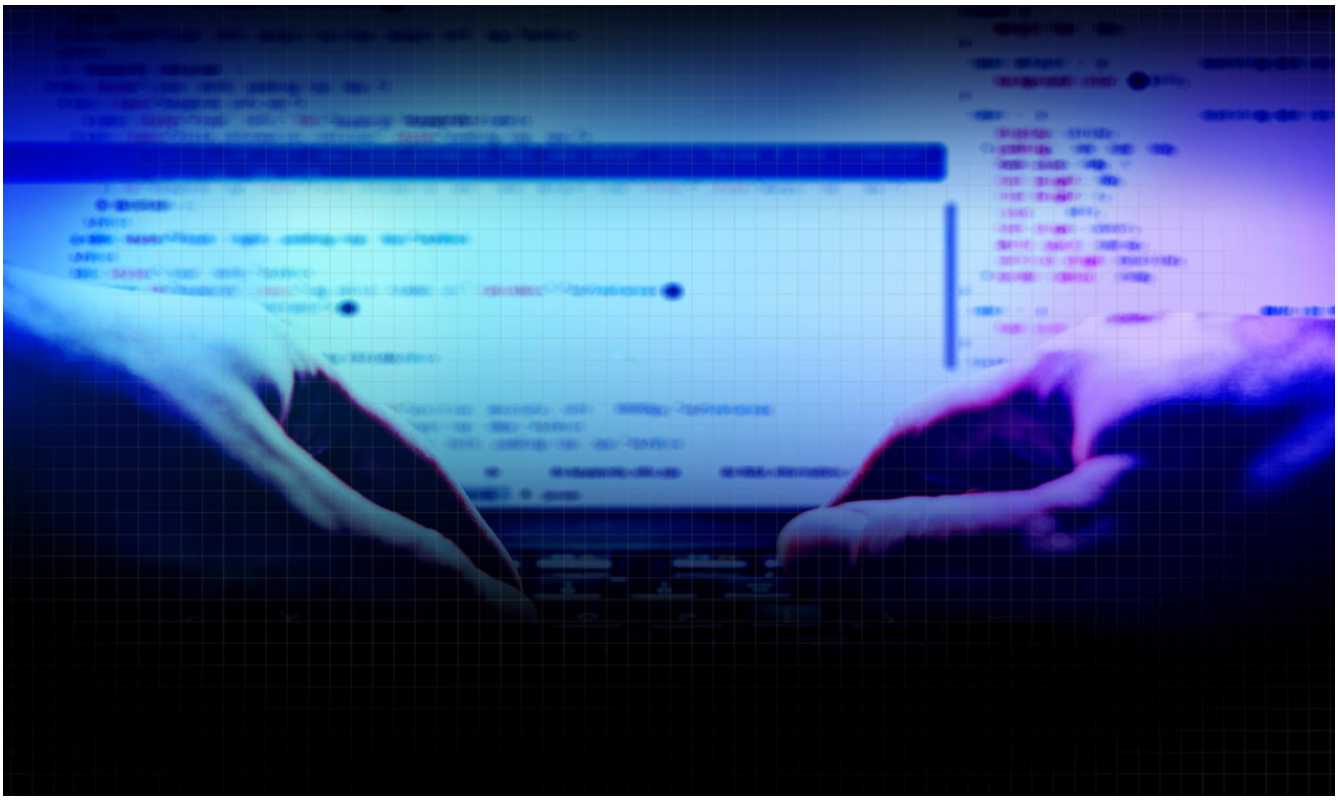
WMI TASKS

Involves files ☑

Possible attack scenarios
executable-less, dual-use, code-injection

Windows Management Instrumentation (WMI) is a powerful set of tools used to manage Windows systems both locally and remotely, and is present on all Windows operating systems.[24] While WMI can be queried to provide much information about the infected system, it is also possible to create a WMI event, which will execute a desired action whenever the terms of the event are met. WMI is covered in more detail in the malicious logic section.

Example: The advanced Poshspy backdoor, deployed by APT29, uses this ability in order to regularly execute its malicious PowerShell payload on the infected host. [25]



THE CHALLENGE

DETECTION AND PROTECTION

Fileless malware has become an increasing threat, with a survey by SANS Institute showing almost one-third of respondents experienced a fileless threat attacking their organization during 2017.[26]

There are several reasons why attackers prefer to use fileless attack methods. For one, the fact that the malicious logic of the attack usually occurs only in memory makes traditional static detection impossible, as no file is saved to disk. In addition, it also complicates post-event analysis, since many artifacts related to the attack exist in memory only, and they might be overwritten or removed by the time of discovery, through a reboot for example. In-memory detection and artifact collection can be done through the use of heuristics and behavioral analysis, which can detect malicious in-memory activities.

Also, the use of scripts and admin tools makes it easy for the attackers to hide their presence and purposes. Scripts can be very easily obfuscated, and delivered in several stages, while actions performed by admin tools might seem legitimate to an organization. The use of scripts and admin tools also makes the sharing of traditional indicators of compromise (IoC's) more difficult, since there are usually no files to share. Instead, organizations should share techniques related to fileless attacks discovered by them.

All the above-mentioned methods enable attackers to hide a large portion of their attacks, which are detected by organizations only post-compromise. Accordingly, a survey conducted in November 2017 by the Ponemon Institute revealed that 50% of respondents wish to augment or have replaced their existing security solutions to deal with fileless attacks. [27]

The security industry has not remained idle, and additional capabilities were added to many security products to combat fileless attacks. A set of capabilities dealing with in-memory analysis, such as dynamic or run-time analysis, and memory heuristics, has been added in order to detect suspicious activities in memory, while security products can also enforce different policies across an organization, which can limit the access some users have to powerful tools such as PowerShell and WMI, or to internal organizational tools. Some security products also contain a compliance capability, which provides administrators with the ability to monitor possible vulnerabilities within their organization – such as unpatched software or use of old operating systems, and enforce updates. Though not perfect, these capabilities can improve the ability to detect fileless attacks.

INSIGHTS AND FORECAST

Fileless malware has gained increasing attention in the past few years due to the publication of many high-profile fileless attacks, including attacks by nation-state actors, and attacks against banks and financial systems. The following are our observations regarding the future of fileless attacks:

- Such attacks are expected to continue and increase in numbers, as they are currently more difficult for security solutions to detect than traditional file-based attacks. However, the increasing use of heuristics and in-memory analysis by security software will make these attacks more difficult, as security programs will be able to stop attacks in memory, and not just scan for malware on disk.
- The increasing sophistication and organization of the cybercrime underground has made sharing of information between cybercriminals much easier.[28] This will contribute not only to the rise in fileless attacks, since information about attack techniques can be shared effectively, but will also increase the sophistication of these attacks, as attackers could research and share new attack methods, such as abuse of WMI, or use of additional script types.
- The abuse of admin tools, or of legitimate internal tools of organizations, will require organizations to change their defense mechanisms to protect themselves from fileless attacks. To be protected, organizations will monitor their network, and have tight control over the users allowed to access administrative or native tools, and over the actions that can be performed by these tools.

THE SOLUTION

RECOMMENDATIONS AND SUMMARY

Regardless of an organizations choice of a security solution, there are some steps organizations and users can take to protect themselves from fileless attacks:

Restrict the use of scripts and scripting languages inside the organization, by applying different policies to different areas of the network. Allow scripts to run from read-only network locations or access only specific machines.

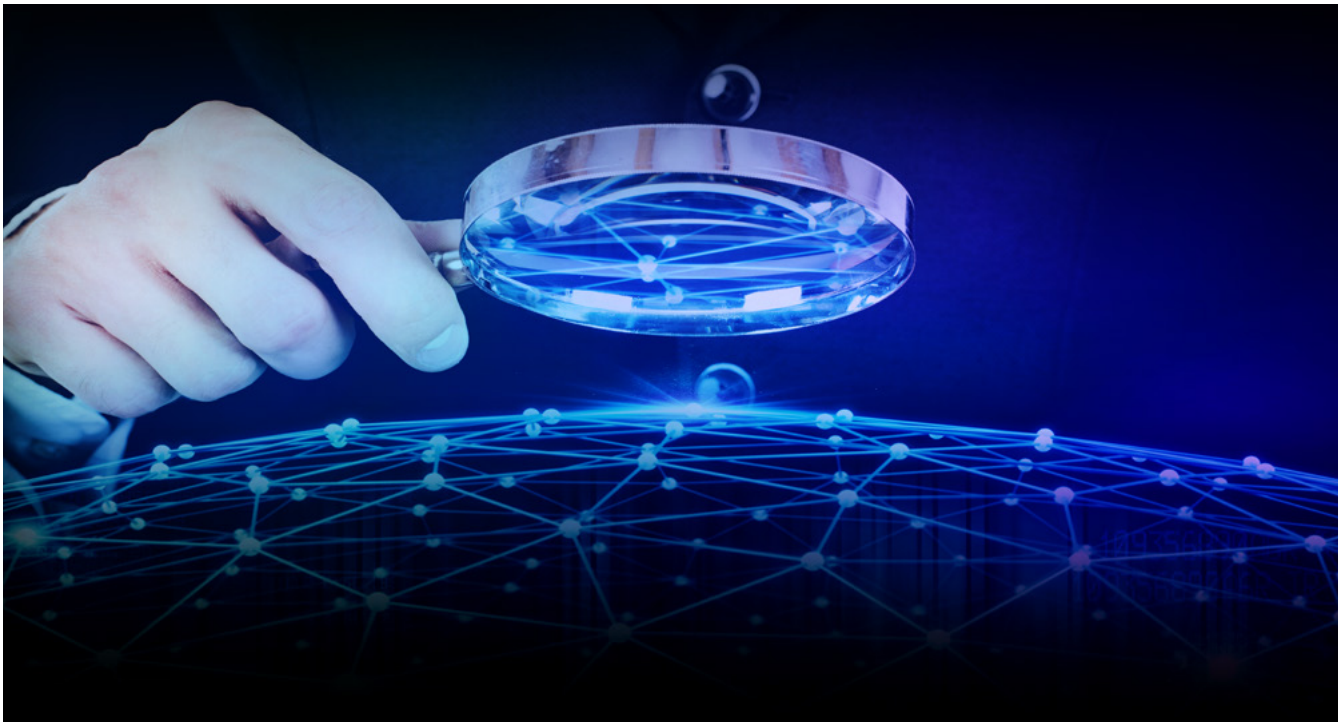
Restrict and monitor the use of Interactive PowerShell and WMI within the organization.

Block execution of macros, and digitally sign trusted macros, which can be allowed to run within the organization.

Make sure all your computers and programs are updated regularly and on time. This will prevent the exploitation of known and patched vulnerabilities.

In any case, **do not click on unknown or untrusted links**, and do not open email attachments which are unknown or untrusted. Infection through social engineering is the most common method of infection.

Deploy an advanced endpoint protection solution which can detect and mitigate fileless attacks. Some advanced endpoint solutions can also enforce all the points mentioned above.



Fileless malware, which can be more accurately termed "executable-less" malware, due to the use of non-executable filetypes during many attack sequences, is a distinct and developing threat. Since malicious actors utilizing fileless malware wish to remain as covert as possible, many unique attack sequences and methods are used in fileless attacks, throughout all stages of the attack. Due to the sophisticated methods of attack, fileless malware attacks currently pose a difficulty for the security industry. However, increasing awareness to these attacks by the security industry are making evasion harder for attackers. Moreover, users and organizations can protect themselves from fileless attacks, to lessen their likelihood of becoming infected. Due to the growth in knowledge of both users and security vendors, malicious actors are expected to increase both the amount of fileless attacks, and their sophistication, and develop new ways through which fileless attacks can be conducted.

Using advanced methodologies and deep learning, Deep Instinct protects its customers from all forms of fileless attacks. Script control mechanisms protect customers from executable-less attacks, code-injection and in-memory attacks, as these attack mechanisms usually rely on scripts during the attack flow. Advanced heuristics, which also protect against file-based attacks, quickly prevent code-injection and in-memory attacks. And finally, Deep Instinct's unique deep learning model protects against dual-use tools utilized in living off the land attacks, and against dropper files used in executable-less attacks, blocking these attacks pre-execution.

BIBLIOGRAPHY

1 https://thebestvpn.com/cyber-security-statistics-2018/

2 https://tinyurl.com/y7gxzrap

3 The first truly fileless malware is the Code Red worm, which appeared in 2001. The worm sent its code as an HTTP request, which exploited a buffer overflow vulnerability on the victim, and ran its code directly in memory. For more information: https://www.sans.org/reading-room/whitepapers/malicious/code-red-worm-45.

4 https://tinyurl.com/y7gxzrap

5 https://www.sans.org/press/announcement/2017/08/07/1

6 https://www.darkreading.com/perimeter/fileless-malware-attacks-hit-milestone-in-2017/d/d-id/1330691

7 https://blog.fortinet.com/2017/11/27/cobalt-malware-strikes-using-cve-2017-11882-rtf-vulnerability

8 http://malware.dontneedcoffee.com/2014/08/angler-ek-now-capable-of-fileless.html

9 http://blog.morphisec.com/iranian-fileless-cyberattack-on-israel-word-vulnerability

10 https://docs.microsoft.com/en-us/powershell/scripting/getting-started/getting-started-with-windows-powershell?view=powershell-5.1

11 https://github.com/PowerShellMafia/PowerSploit

12 https://msdn.microsoft.com/en-us/library/aa394582(v=vs.85).aspx

13 https://www.blackhat.com/docs/us-15/materials/us-15-Graeber-Abusing-Windows-Management-Instrumentation-WMI-To-Build-A-Persistent%20Asynchronous-And-Fileless-Backdoor-wp.pdf

14 https://www.fireeye.com/blog/threat-research/2017/03/dissecting_one_ofap.html

15 https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-living-off-the-land-and-fileless-attack-techniques-en.pdf

16 https://threatpost.com/fileless-memory-based-malware-plagues-140-banks-enterprises/123652/

17 https://www.bleepingcomputer.com/news/security/-process-doppelg-nging-attack-works-on-all-windows-versions/

18 https://reaqta.com/2015/09/poweliks-file-less-malware-keeps-evolving/

19 https://reaqta.com/2015/09/poweliks-file-less-malware-keeps-evolving/

20 https://msdn.microsoft.com/library/ms694363.aspx

21 https://attack.mitre.org/wiki/Technique/T1122

22 https://www.gdatasoftware.com/blog/2014/10/23941-com-object-hijacking-the-discreet-way-of-persistence

23 http://blog.morphisec.com/iranian-fileless-cyberattack-on-israel-word-vulnerability

24 https://www.blackhat.com/docs/us-15/materials/us-15-Graeber-Abusing-Windows-Management-Instrumentation-WMI-To-Build-A-Persistent%20Asynchronous-And-Fileless-Backdoor-wp.pdf

25 https://www.fireeye.com/blog/threat-research/2017/03/dissecting_one_ofap.html

26 https://www.sans.org/press/announcement/2017/08/07/1

27 https://tinyurl.com/y7gxzrap

28 https://www.europol.europa.eu/sites/default/files/documents/iocta2017.pdf

ABOUT DEEP INSTINCT

Deep Instinct is an omni-cybersecurity platform that helps companies and organizations protect themselves against zero-day, APT and ransomware attacks with unmatched accuracy. By providing deep learning predictive capabilities, and a solution that is based on a proprietary deep learning framework Deep Instinct is revolutionizing cybersecurity. Deep Instinct's solution provides comprehensive defense designed to protect against known and unknown malware in real-time, across endpoints, servers, and mobile devices. Deep learning's capabilities of identifying malware from any data source results in comprehensive protection on any device and operating system.

To learn more about Deep Instinct capabilities, get a personal demo from one of our experts

GET A DEMO

deepinstinct
BEFORE YOU KNOW IT

www.deepinstinct.com

© Deep Instinct Ltd. This document contains proprietary information. Unauthorized use, duplication, disclosure or modification of this document in whole or in part without written consent of Deep Instinct Ltd.. is strictly prohibited. Deep Instinct has invested significant efforts to make this research as updated as possible.